```java
package socialmedia;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import java.io.File;

import java.lang.StringBuilder;

/**
 * BadSocialMedia is a minimally compiling, but non-functioning implementor of
 * the SocialMediaPlatform interface.
 *
 * @author Diogo Pacheco
 * @version 1.0
 */
public class SocialMedia implements SocialMediaPlatform {

    ArrayList<Account> accounts;
    ArrayList<Post> posts;

    /**
     * constructor
     */
    public SocialMedia(){
        accounts = new ArrayList<Account>();
        posts = new ArrayList<Post>();
    }

    @Override
    public int createAccount(String handle) throws IllegalHandleException,
    InvalidHandleException {
        // TODO Auto-generated method stub
        if (handleExists(handle)){
            throw new IllegalHandleException();
        }
        if (!isValidHandle(handle)){
            throw new InvalidHandleException();
        }
        Account newAccount = new Account(handle);
        accounts.add(newAccount);
        return newAccount.getID();
    }

    @Override
    public int createAccount(String handle, String description) throws
    IllegalHandleException, InvalidHandleException {
        // TODO Auto-generated method stub
        Account newAccount = new Account(handle, description);
        accounts.add(newAccount);
        return newAccount.getID();
    }

    @Override
    public void removeAccount(int id) throws AccountIDNotRecognisedException {
        // TODO Auto-generated method stub
        for (Account a : accounts){
            if (a.getID() == id){
                accounts.remove(a);
                return;
            }
        }
        throw new AccountIDNotRecognisedException();
    }

    @Override
    public void removeAccount(String handle) throws HandleNotRecognisedException {
        // TODO Auto-generated method stub
        accounts.remove(getAccountByHandle(handle));
```

```
71              throw new HandleNotRecognisedException();
72          }
73
74          @Override
75          public void changeAccountHandle(String oldHandle, String newHandle)
76                  throws HandleNotRecognisedException, IllegalHandleException,
                    InvalidHandleException {
77              // TODO Auto-generated method stub
78              if (handleExists(newHandle)){
79                  throw new IllegalHandleException();
80              }
81              if (!isValidHandle(newHandle)){
82                  throw new InvalidHandleException();
83              }
84              Account accountToChange = getAccountByHandle(oldHandle);
85              if (accountToChange == null){
86                  throw new HandleNotRecognisedException();
87              }
88
89              accountToChange.updateHandle(newHandle);
90
91          }
92
93          @Override
94          public void updateAccountDescription(String handle, String description) throws
            HandleNotRecognisedException {
95              // TODO Auto-generated method stub
96              Account accountToChange = getAccountByHandle(handle);
97              if (accountToChange == null){
98                  throw new HandleNotRecognisedException();
99              }
100
101             accountToChange.updateDescription(handle);
102
103         }
104
105         @Override
106         public String showAccount(String handle) throws HandleNotRecognisedException {
107             // TODO Auto-generated method stub
108             Account account = getAccountByHandle(handle);
109             if (account == null){
110                 throw new HandleNotRecognisedException();
111             }
112             return account.showAccount();
113         }
114
115         @Override
116         public int createPost(String handle, String message) throws
            HandleNotRecognisedException, InvalidPostException {
117             // TODO Auto-generated method stub
118             Account poster = getAccountByHandle(handle);
119             if (poster == null){
120                 throw new HandleNotRecognisedException();
121             }
122             if(!isValidPost(message)){
123                 throw new InvalidPostException();
124             }
125             Post newPost = new Original(poster, message);
126             posts.add(newPost);
127             poster.addPost(newPost);
128             return newPost.getID();
129         }
130
131         @Override
132         public int endorsePost(String handle, int id)
133                 throws HandleNotRecognisedException, PostIDNotRecognisedException,
                    NotActionablePostException {
134             // TODO Auto-generated method stub
135             // EP@exampleuser:Endorsed message
136             Account endorser = getAccountByHandle(handle);
137             if (endorser == null){
138                 throw new HandleNotRecognisedException();
```

```
139             }
140             Post postToEndorse = getPostByID(id);
141             if(postToEndorse == null){
142                 throw new PostIDNotRecognisedException();
143             }
144             if (postToEndorse instanceof Endorsement || postToEndorse instanceof
                EmptyPost){
145                 throw new NotActionablePostException();
146             }
147             Endorsement newEndorsement = new Endorsement(endorser, postToEndorse);
148             postToEndorse.addEndorsements(newEndorsement);
149             posts.add(newEndorsement);
150             endorser.addPost(newEndorsement);
151             getAccountByHandle(postToEndorse.getAccountHandle()).increaseEndorsements();
152             return newEndorsement.getID();
153         }
154 
155         @Override
156         public int commentPost(String handle, int id, String message) throws
                HandleNotRecognisedException,
157                 PostIDNotRecognisedException, NotActionablePostException,
                    InvalidPostException {
158             // TODO Auto-generated method stub
159             Account commenter = getAccountByHandle(handle);
160             if(commenter == null){
161                 throw new HandleNotRecognisedException();
162             }
163             Post postToComment = getPostByID(id);
164             if(postToComment == null){
165                 throw new PostIDNotRecognisedException();
166             }
167             if (postToComment instanceof Endorsement || postToComment instanceof
                EmptyPost){
168                 throw new NotActionablePostException();
169             }
170             if(!isValidPost(message)){
171                 throw new InvalidPostException();
172             }
173             Comment newComment = new Comment(commenter, message, postToComment);
174             posts.add(newComment);
175             postToComment.addComment(newComment);
176             commenter.addPost(newComment);
177             return newComment.getID();
178         }
179 
180         @Override
181         public void deletePost(int id) throws PostIDNotRecognisedException {
182             // TODO Auto-generated method stub
183             Post postToRemove = getPostByID(id);
184             if (postToRemove == null){
185                 throw new PostIDNotRecognisedException();
186             }
187             posts.removeAll(postToRemove.getEndorsements());
188             postToRemove.deleteEndorsements();
189             int index = posts.indexOf(postToRemove);
190             EmptyPost newEmptyPost = new EmptyPost(postToRemove);
191             posts.set(index, newEmptyPost);
192             getAccountByHandle(postToRemove.getAccountHandle()).removePost(postToRemove);
193             ArrayList<Comment> oldComments = postToRemove.getComments();
194             for (Comment c : oldComments){
195                 c.setOriginalPost(newEmptyPost);
196             }
197         }
198 
199         @Override
200         public String showIndividualPost(int id) throws PostIDNotRecognisedException {
201             // TODO Auto-generated method stub
202             Post postToShow = getPostByID(id);
203             if (postToShow == null){
204                 throw new PostIDNotRecognisedException();
205             }
206             return postToShow.show(0);
```

```
207         }
208 
209         @Override
210         public StringBuilder showPostChildrenDetails(int id)
211                 throws PostIDNotRecognisedException, NotActionablePostException {
212             // TODO Auto-generated method stub
213             Post postToShow = getPostByID(id);
214             if (postToShow == null){
215                 throw new PostIDNotRecognisedException();
216             }
217             if (postToShow instanceof Endorsement){
218                 throw new NotActionablePostException();
219             }
220             StringBuilder sb = new StringBuilder(postToShow.showWithChildren(0));
221             return sb;
222         }
223 
224         @Override
225         public int getNumberOfAccounts() {
226             // TODO Auto-generated method stub
227             return accounts.size();
228         }
229 
230         @Override
231         public int getTotalOriginalPosts() {
232             // TODO Auto-generated method stub
233             int numOriginalPosts = 0;
234             for (Post p : posts){
235                 if (p instanceof Original){
236                     numOriginalPosts++;
237                 }
238             }
239             return numOriginalPosts;
240         }
241 
242         @Override
243         public int getTotalEndorsmentPosts() {
244             // TODO Auto-generated method stub
245             int numEndorsementPosts = 0;
246             for (Post p : posts){
247                 if (p instanceof Endorsement){
248                     numEndorsementPosts++;
249                 }
250             }
251             return numEndorsementPosts;
252         }
253 
254         @Override
255         public int getTotalCommentPosts() {
256             // TODO Auto-generated method stub
257             int numCommentsPosts = 0;
258             for (Post p : posts){
259                 if (p instanceof Comment){
260                     numCommentsPosts++;
261                 }
262             }
263             return numCommentsPosts;
264         }
265 
266         @Override
267         public int getMostEndorsedPost() {
268             // TODO Auto-generated method stub
269             Post mostEndorsed = null;
270             boolean firstFound = false;
271             for (Post p : posts){
272                 if (p instanceof Original || p instanceof Comment){
273                     if (firstFound){
274                         if (p.getNumEndorsements() > mostEndorsed.getNumEndorsements()){
275                             mostEndorsed = p;
276                         }
277                     }
278                     else{
```

```
                    mostEndorsed = p;
                }
            }
        }
        return mostEndorsed.getID();
    }

    @Override
    public int getMostEndorsedAccount() {
        // TODO Auto-generated method stub
        Account mostEndorsed = null;
        boolean firstFound = false;
        for (Account a : accounts){
            if (firstFound){
                if (a.getEndorseCount() > mostEndorsed.getEndorseCount()){
                    mostEndorsed = a;
                }
            }
            else{
                mostEndorsed = a;
            }
        }
        return mostEndorsed.getID();
    }

    @Override
    public void erasePlatform() {
        // TODO Auto-generated method stub
        posts.clear();
        accounts.clear();

    }

    @Override
    public void savePlatform(String filename) throws IOException {
        // TODO Auto-generated method stub
        try{
            Object arr[] = new Object[2];
            arr[0] = accounts;
            arr[1] = posts;
            FileOutputStream fileOut = new FileOutputStream( filename + ".obj");
            ObjectOutputStream objOut = new ObjectOutputStream(fileOut);
            objOut.writeObject(arr);
            objOut.close();
        }
        catch(IOException e){
            throw e;
        }
        catch (Exception e){
            System.out.println("something went wrong, man");
        }

    }

    @Override
    public void loadPlatform(String filename) throws IOException,
    ClassNotFoundException {
        // TODO Auto-generated method stub
        try{
            FileInputStream inStream = new FileInputStream(filename + ".obj");
            ObjectInputStream objIn = new ObjectInputStream(inStream);
            Object arr[] = (Object[])objIn.readObject();
            accounts = (ArrayList<Account>)arr[0];
            posts = (ArrayList<Post>)arr[1];
            objIn.close();
        }
        catch(IOException e){
            throw e;
        }
        catch(ClassNotFoundException e){
            throw e;
        }
```

```
350             catch(Exception e){
351                 System.out.println("something went wrong, man");
352             }
353 
354         }
355 
356         private boolean handleExists(String handle){
357             for (Account a : accounts){
358                 if (a.getHandle().equals(handle)){
359                     return true;
360                 }
361             }
362             return false;
363         }
364 
365         private boolean isValidHandle(String handle){
366             //no whitespace, no empty, no more than 30 characters
367             if (handle.isEmpty()){
368                 return false;
369             }
370             if (handle.contains(" ")){
371                 return false;
372             }
373             if (handle.length() > 30){
374                 return false;
375             }
376             return true;
377         }
378 
379         private boolean isValidPost(String content){
380             if(content.isEmpty()){
381                 return false;
382             }
383             if (content.length() > 100){
384                 return false;
385             }
386             return true;
387         }
388 
389         private Account getAccountByHandle(String handle){
390             for (Account a : accounts){
391                 if (a.getHandle().equals(handle)){
392                     return a;
393                 }
394             }
395             return null;
396         }
397 
398         private Post getPostByID(int id){
399             for (Post p : posts){
400                 if (p.getID() == id){
401                     return p;
402                 }
403             }
404             return null;
405         }
406     }
407 
```